

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 04-100148  
(43)Date of publication of application : 02.04.1992

(51)Int.Cl. G06F 9/06  
G06F 15/00

(21)Application number : 02-263242 (71)Applicant : SUN MICROSYST INC  
(22)Date of filing : 02.10.1990 (72)Inventor : CORBIN JOHN R

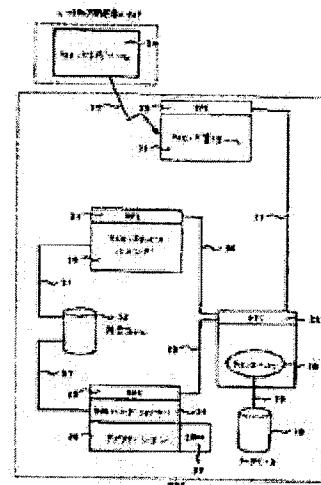
(30)Priority

Priority number : 89 415984 Priority date : 02.10.1989 Priority country : US

## (54) METHOD FOR PREVENTING UNAUTHORIZED USE OF SOFTWARE INSIDE COMPUTER NETWORK SYSTEM

### (57)Abstract:

**PURPOSE:** To prevent unauthorized use of a software application by confirming a compounded license token means before the certification of access to a license. **CONSTITUTION:** When a user desires the execution of the software application, a licensing library 24 performs calling for requesting a license token from a license server 20. The license server 20 detects the proper license token of the software application and transmits this license token to the licensing library 24, and a license access module connected to the licensing library 24 the license token. A routine inside the licensing library coupled to the software application 26 barrows the license and confirms license information before the license token is updated. Thus, the illegal use of the software application is prevented.



## ⑫ 公開特許公報 (A) 平4-100148

⑬ Int. Cl.<sup>5</sup>G 06 F 9/06  
15/00

識別記号

450 P 7927-5B  
330 B 7218-5L

府内整理番号

⑬ 公開 平成4年(1992)4月2日

審査請求 未請求 請求項の数 2 (全17頁)

⑭ 発明の名称 コンピュータネットワークシステム内でのソフトウェアの不正使用  
を防ぐ方法

⑭ 特 願 平2-263242

⑭ 出 願 平2(1990)10月2日

優先権主張 ⑭ 1989年10月2日 ⑭ 米国(US)⑭ 415,984

⑮ 発明者 ジョン・リチャード・  
コーピン アメリカ合衆国 94086 カリフォルニア州・サンノゼ  
イリ・アメリカ アヴェニュウ・372

⑮ 出願人 サン・マイクロシステムズ・インコーポレー  
テッド アメリカ合衆国 94043 カリフォルニア州・マウンテン  
ビュー・ガルシア・アヴェニュウ・2550

⑮ 代理人 弁理士 山川 政樹 外3名

## 明細書

## 1. 発明の名称

コンピュータネットワークシステム内でのソフ  
トウェアの不正使用を防ぐ方法

## 2. 特許請求の範囲

(1) 少なくとも1個のネットワークを構成する複数の端末上で作動すべくライセンスされた複数のソフトウェアアプリケーションを含むコンピュータネットワークの環境であって、前記アプリケーションは前記端末上にあり、特定の端末におけるアプリケーションの使用はライセンスの付与によって認可され、前記ライセンスは前記アプリケーションの存在する前記端末からユーザによって要求されることを特徴とするコンピュータネットワークの環境において、前記アプリケーションの不正使用を防ぐシステムであって、

前記アプリケーションのライセンシング情報を記憶するライセンストークン手段と、

前記端末に接続されて前記アプリケーションと

通信を行なうライセンスサーバ手段であって、前記ライセンストークン手段を記憶するデータベースを有し、前記アプリケーションによるライセンスへの要求に応じて前記データベースから前記ライセンストークン手段を検索し、さらに前記ライセンストークン手段を前記アプリケーションへと伝達するライセンスサーバ手段と、

前記端末に接続され、前記ライセンスサーバ手段から伝達されるライセンストークン手段の符号化と復号化とを行なうライセンスマネジメント手段であって、前記アプリケーションと一緒に化され、前記ライセンストークン手段を前記ライセンスサーバ手段から受信するライセンスマネジメント手段と、

前記端末に接続され、前記ライセンスへのアクセスが認可される前に前記復号化されたライセンストークン手段を確認するライセンシングライブライアリ手段であって、前記アプリケーションと一緒に化されたライセンシングライブライアリ手段と、

からなることを特徴とする前記アプリケーションの不正使用を防ぐシステム。

(2) 少なくとも1個のネットワークを構成する複数の端末上で作動すべくライセンスされた複数のソフトウェアアプリケーションを含むコンピュータネットワークの環境であって、前記アプリケーションは前記端末上にあり、特定の端末におけるアプリケーションの使用はライセンスの付与によって認められ、前記ライセンスは前記アプリケーションの存在する前記端末からユーザによって要求されることを特徴とするコンピュータネットワークの環境において、前記アプリケーションの不正使用を防ぐシステムであって、

前記アプリケーションのライセンシング情報を記憶するライセンストークン手段と、

前記端末に接続されて前記アプリケーションと通信を行なうライセンスサーバ手段であって、前記ライセンストークン手段を記憶するデータベースを有し、前記アプリケーションによるライセンスへの要求に応じて前記データベースから前記ライセンストークン手段を検索し、さらに前記ライセンストークン手段を前記アプリケーションへと

ス結合手段と、  
からなることを特徴とする前記アプリケーションの不正使用を防ぐシステム。

### 3.発明の詳細な説明

#### 【産業上の利用分野】

本発明は、コンピュータネットワーク環境におけるソフトウェアアプリケーションの不正使用を防ぐ方法に関する。

#### 【従来技術】

コンピュータネットワークとは、一般にリンクやケーブルで相互に接続された複数のマシンまたは端末をいう。コンピュータネットワークは、アクセスに対してオープンであるという特徴を有するため、ソフトウェアを許可なく複製することが可能である。このためソフトウェア開発者のライセンシング収入が減少する結果となっている。従来、ライセンスによってネットワーク全体の実施の許諾を行なう(一般にサイトライセンスと呼ば

伝達するライセンスサーバ手段と、

前記アプリケーションに接続され、前記端末からアクセス可能であり、前記ライセンスサーバ手段から伝達されるライセンストークン手段の符号化と復号化とを行なうライセンスアクセス手段であって、前記アプリケーションと一体化されたライセンスアクセス手段と、

前記アプリケーションに接続され、前記端末からアクセス可能であり、前記ライセンスへのアクセスが許可される前に前記復号化されたライセンストークン手段を確認するライセンシングライブラリ手段であって、前記アプリケーションと一体化されたライセンシングライブラリ手段と、

前記ライセンスサーバ手段と前記ライセンシングライブラリ手段とに接続され、結合ファイルを形成するライセンス結合手段であって、前記結合ファイルは前記ライセンシングライブラリ手段に対して前記ライセンスサーバ手段のうち、いずれの手段が前記アプリケーションにライセンスを付与できるかを通知することを特徴とするライセン

れる)か、もしくはソフトウェアが実行される各ノードをライセンスする(一般にノードライセンスと呼ばれる)必要があった。ノードとは、コンピュータネットワーク内の各マシン、端末もしくはシステムを指す。ライセンスとは、ソフトウェアの開発者が顧客に対して任意のソフトウェアアプリケーションを特定の様態で使用することを許諾することをいう。

サイトライセンスの場合、特定の場所すなわちネットワーク上の全ユーザに、各々のネットワーク上の位置に關係なくソフトウェアアプリケーションの使用を許可する。均一の使用料を徴収するこの方法は、使用頻度の低いソフトウェアアプリケーションの場合には行き過ぎとなる。ノードライセンスの場合、ソフトウェアアプリケーションをネットワーク上の特定のマシンに結びつけてしまうことになり、またソフトウェアアプリケーションの使用頻度が低ければ課算がとれない。これらの点については米国特許4,688,169に開示されている。またライセンスを付与されたノードで新

たなユーザがソフトウェアアプリケーションを使用しようとする場合、こうしたユーザはライセンスの新規購入を求められることが多い。

サイトライセンスやノードライセンスに代わる概念として並行使用ライセンスがある。並行使用ライセンスでは、任意の時点でのソフトウェアアプリケーションの使用を許可されるユーザの台数を、ネットワーク上の位置に係わりなく限定している。ビデオ愛好者がレンタルビデオ店から映画のビデオを借り出すように、ネットワーク上のユーザはソフトウェアアプリケーションを端末から先着順に借り出す。このように、並行使用ライセンスでは、ソフトウェアアプリケーションの実際の使用度に比例して使用料が徴収される。

ネットワーク環境でソフトウェアアプリケーションの並行使用を許可するというライセンシング方法は、現在ハイランド・ソフトウェア社とアポロ・コンピュータ社で採用されている。これに関する書籍は、「並行アクセスライセンシング(Concurrent Access Licensing)」(1988年9月、Vol. 6、

No. 8、ユニックス・レビュー(Unix Review)、H. オルソン、P. ルヴァイン(H. Olson and P. Levin e)著)で述べられている。一般に、ソフトウェアアプリケーションのライセンスは、ライセンスサーバの制御するデータベースに記憶されている。ライセンスサーバとは、ライセンスを記憶しているプログラムであり、かつライセンスを貸出す前にユーザの資格を確認するプログラムである。不正使用を防ぐために並行使用をライセンスするこれらの方針は、公共／個人キー暗号化方式などによって保護された通信手段に依存している。かかる公共／個人キー暗号化方式によれば、システムの各ユーザは2個のキーを有する。そのうち1個のキーは通常公共のキーであり、他の1個は個人用のキーである。個人キーを使用した個人変換は、公共キーを使用した公共変換に関連付けられているが、個人キーは公共キーから計算によって求めることはできない。これらの点については、D. デニング(D. Denning)著「暗号化技術とデータ保護(Cryptography and Data Security)」(1982年、

アディソン・ウェスリー(Addison-Wesley)社刊)に述べられている。暗号キーはライセンスサーバ内に隠されていて、ライセンスのデータベースの暗号化に使用される。高度な技術で設計された公共／個人キー暗号化方式を解説することは難しい。とくにライセンスサーバが安全な環境内にあれば、かかる方式の解説は至難の技である。安全な環境とは、そのアクセスが資格を有するユーザのみに限定されている環境をいう。しかしながら、ライセンスサーバは顧客のサイト、すなわち危険な環境に設置されることが多い。したがって、ライセンスサーバは熟練したハッカーの解説にさらされことになる。いったん個人キーが解説されると、ライセンスサーバに関するあらゆる重要な情報(ライセンス等)が無許可のまま公開されてしまう。

したがって、本発明の目的は、並行使用ライセンシングの環境におけるソフトウェアの不正使用を防ぐためのより安全性な方法を提供することにある。

#### 【発明の概要】

本発明は、通常ライセンスサーバが行なっていた確認機能とライセンス貸出機能とをソフトウェアアプリケーションに与えるものである。本発明の好ましい実施例は、少なくとも1個のライセンスサーバと少なくとも1個のソフトウェアアプリケーションとを実行する複数の端末を含むコンピュータネットワークからなる。ライセンスサーバは、ソフトウェアアプリケーションのライセンス情報を格納している端末のデータベースを制御する。ライセンス情報はライセンストークンに含まれており、またライセンスサーバが制御するデータベースに記憶されている。ライセンストークンは、特定のビットパターンすなわちパケットであって、アプリケーションソフトウェアの販売業者によって暗号化されている。ソフトウェアアプリケーションは、ライセンシングライブラリを介してライセンスサーバと通信を行なう。ライセンシングライブラリはライブラリルーチンの集合であって、ソフトウェアアプリケーションがライセン

スをライセンスサーバから要求する場合か、もしくは該ライセンスを更新する場合にこれらのライセンスリルーチンを使用する。ライセンスアクセスモジュールは、ソフトウェアアプリケーションとライセンシングライブライアリとに接続しているプログラムであって、ライセンストークンを販売業者に特定のフォーマットからライセンシングライブライアリのフォーマットへと復号化するプログラムである。

ユーザがソフトウェアアプリケーションの実行を希望する場合、ライセンシングライブライアリは、ライセンストークンをライセンスサーバから要求するための呼び出しを行なう。従来、ライセンスサーバはユーザの資格を確認したのち該要求を許可または却下していたが、本実施例のライセンスサーバは、当該ソフトウェアアプリケーションの正しいライセンストークンを検出してこれをライセンシングライブライアリに伝達する。ライセンシングライブライアリに接続されたライセンスアクセスモジュールは該ライセンストークンを復号化する。ソ

アクセスモジュールとを発表しているはずである。このように、本発明は、基本的なコンピュータネットワークを変更することなく、コンピュータネットワーク環境におけるソフトウェアアプリケーションの不正使用を防ぐためのより安全な方法を提供する。

#### 【表記と名称】

以下の詳細な説明では、データビットやコンピュータメモリ内のデータ構造に対して行なわれる演算のアルゴリズムや記号表記を主として使用する。アルゴリズムに基づくこれらの記述や表記は、データ処理分野の有能な技能者が、各々の作業の内容を他の同業者に最も効果的に伝達するために使用する手段である。

以下でいうアルゴリズムとは、通常定義されるように、所望の結果を得るための一貫性のある一連のステップをいう。これらのステップは、物理量の物理的な操作を必要とするステップである。これらの物理量は、必ずしも限定されないが、電

ソフトウェアアプリケーションに結合されたライセンシングライブライアリ内のルーチンは、ライセンスを借り出してライセンストークンを更新する前に、ライセンス情報を確認する。ライセンスアクセスモジュールは、更新されたライセンストークンを符号化したのち、該トークンをライセンスサーバへと返送する。

ライセンストークンの確認機能と借り出し機能とがソフトウェアアプリケーションによって実行されるため、ライセンスサーバに代わってソフトウェアアプリケーションが無許可ユーザの解説の対象となる。ライセンスアクセスモジュールにリバースエンジニアリングをかけても、ライセンスサーバの解説より実りが少ない。この理由は、ライセンスアクセスモジュールはライセンスデータベースのごく一部の内容を開示するに過ぎないからである。大部分のハッカーがライセンスアクセスモジュールを解説する頃には、ソフトウェア販売業者は当該ソフトウェアアプリケーションの新バージョンと、これに対応する新たなライセンス

的もしくは強制的な信号の形態をとることが普通であり、かかる信号に対しては記憶、転送、組合せ、比較その他の操作を行なうことができる。主として慣例上の理由で、これらの信号をビットパターン、値、要素、記号、文字、データパッケージ等の名称で呼ぶことが便利であることが多い。しかしながら、これらの用語もしくはこれに準ずる用語は、適切な物理量に対応していないばならず、かつ該用語はこれらの数量に付加された便利なラベルに過ぎないことを示知しておく必要がある。

さらに、これらの物理量に対する操作は、加算や比較といった用語で表現され、こうした用語は人間が自分自身で行なう演算に通常関連付けられている。しかしながら、以下で説明する本発明の一部をなす演算では、人間のかかる能力はほとんどの場合不足であり、むしろ好ましくない。該算はマシンが実行するからである。本発明は、汎用ディジタルコンピュータもしくは同様の装置に適用して効果的である。いずれの場合でも、コンピ

ピュータを操作する方法と、計算自体の方法とを明確に区別する必要がある。本発明は、電気信号もしくは他の物理的（機械的、化学的など）信号を処理するためにコンピュータを操作することによって、他の所望の物理的信号を生成するステップの方式に関する。

また本発明は、上記の操作を実行する装置に関する。かかる装置は、所望の目的のためのみに形成されるか、もしくは汎用コンピュータからなり、該コンピュータ内に記憶されたコンピュータプログラムによって選択的に起動され、再構成される。本明細書に含まれるアルゴリズムは、特定のコンピュータやその他の装置に限定されるものではない。また各種の汎用マシンを、本明細書記載の命令に準じて記述されたプログラムとともに使用してもよい。所定のステップを実行するための専用の装置を形成することが便利である場合もある。これらの各種マシンの所定の構造は、以下の説明から明瞭になるであろう。

支えない。好ましい端末の例としては、サン・マイクロシステムズ(Sun Microsystems)社（在カリフォルニア州マウンテンビュー(Mountain View)）の製造するマシンがある。各端末は、キーボード11、11'、11''またはマウス12、12'、12''などの入力装置を有する。第1図に示されるように、端末10 - 10''（図中では10、10'、10''）は共通ケーブル13によって接続され、相互にデータの転送を行なう。共通ケーブル13は、同軸ケーブル、光ケーブル、無線チャネルその他のいかなる共有メディアでもよいことは当業者には明白であろう。さらにケーブル13と端末10 - 10''（図中では10、11'、11''）からなるネットワークは、現状結線、星形結線、バス結線など多種の形態のうちいずれをもとることができ、またゲートウェイやブリッジで結合された複数の小形のネットワークの集合を含んでいてもよい。

再び第1図において、14はライセンスサービスを示す。ライセンスサービス14は、ネットワークに接続されだすべての端末が共有する資源である。

#### 【発明の実施例】

以下の説明は複数の節に分割されている。最初の節では、ライセンスされたソフトウェアプログラムのデータベースをアクセスするための汎用ネットワーク環境を説明する。続く節では、ソフトウェアアプリケーションの不正使用を防ぐ方法の詳細を説明する。

#### 1. 汎用ネットワーク環境

第1図において、コンピュータネットワーク環境は、複数のデータ処理装置10 - 10''（図中では10、10'、10''）からなる。これらのデータ処理装置には、端末、パソコン、ワークステーション、ミニコン、メインフレームコンピュータ、スーパーコンピュータ等が含まれる。本明細書では、本発明のネットワークに接続されたすべてのデータ処理装置を「端末」と称する。これらの端末の製造元は異なるものであって差支えない。また該端末の使用するオペレーティングシステムは、MS-DOS、UNIX、OS/2、MAC OS等さまざまであって差

本実施例では、ライセンスサービス14にはライセンスサーバ15 - 15''（図中では15、15'、15''）とデータベース17 - 17''（図中では17、17'、17''）が含まれる。ここで、nは0以下の値である。ライセンスサーバは、メモリ型記憶機能を有する端末上で実行されるプログラムである。各ライセンスサーバ15（図中では15、15'、15''）は、インタフェース16（図中では16、16'、16''）を介して、端末のメモリに記憶されたデータベース17と通信を行なう。以下で詳しく説明するように、データベース17は、各種のソフトウェアアプリケーションのライセンシング情報を記憶している。これらのソフトウェアアプリケーションは、購入後、コンピュータネットワーク環境で実行するための許可を付与される。ライセンスサーバは、特定の端末での実行に限定されるわけではなく、いずれの端末でも実行可能であり、ユーザがアプリケーションを使用する端末でも実行できる。このように、ネットワークに接続された端末であれば、ユーザがアプリケーションソフトウェアを実行する端末

としてだけでなく、ライセンスサーバとしても機能できる。以下で詳しく説明するように、ライセンスサーバはアプリケーションソフトウェアのライセンスの確認を行なわない。ライセンスサーバの作動形態は受動的であって、記憶機能、ロック機能、ログ機能およびクラッシュ回復機能をアプリケーションソフトウェアに提供している。

第2図は、本発明に基づくネットワークライセンシング方式の構造を示す。この構造には、データベース18、データベースインタフェース19、ライセンスサーバ20、ライセンシングライブラリ24、ライセンスアクセスモジュール27、ライセンス管理ツール21、ライセンスサービスバインダ29、ライセンス生成ツール34が含まれる。

データベース18は、ライセンシング情報とアプリケーション使用データを記録している。データベース18には、以下の情報を含む複数のレコードが格納されていることが望ましい。

データベースの要素	説明
・一意キー テーブル	他の全テーブル用キー
・販売業者 テーブル	販売業者の識別記号
・製品 テーブル	(ID)と名前
・バージョン テーブル	製品の番号と名前
・ライセンス テーブル	バージョンの番号と日付
・ライセンストークン テーブル	ライセンス番号、失效日、総数
・単位 グループ テーブル	符号化されたライセンス
・グループ リスト テーブル	ストークンを格納するテーブル
・使用許可ユーザの テーブル	任意のグループ内のライセンスの配分
・現行ライセンス 使用 テーブル	グループの名前
	使用許可ユーザの資格
	ライセンスを使用中のアプリケーション

データベースの要素	説明
・ロック テーブル	データベース中のロックされたレコード
・有資格管理者 テーブル	管理者のログイン名
・ライセンス操作 ログ テーブル	管理者のログ情報
・ライセンス 使用 ログ テーブル	要求操作とクライアントログ
・ライセンス 持ち行 列 ログ テーブル	ライセンス持ち行列
・アプリケーション メッセージ ログ テーブル	アプリケーションに固有なメッセージ

データベースインタフェース19は、ライセンスサーバ20とデータベース18との間の通信を提供する。これによって複数のユーザがデータベース内の同一のレコードに同時にアクセスすることを防いでいる。かかる同時にアクセスが行なわれると、該レコード内のデータが破壊されることがある。

このように、アプリケーションの使用時にはロックの所有者のみがロックされたレコードからデータを読み出し、書き込むことができる。

ライセンスサーバ20は端末上で作動し、データベース18と、ライセンス管理ツール21、ライセンシングライブラリ24およびライセンスサービスバインダ29とのインタフェースをとる。ライセンスサーバ20は、インターフェース23を介して、ライセンス管理ツール21、ライセンシングライブラリ24およびライセンスサービスバインダ29と通信を行なう。インターフェース23は、遠隔手順呼出し機構であることが望ましい。遠隔手順呼出し機構によれば、ネットワークに接続された1個の装置または端末上で作動するプロセスは、該ネットワークに接続された遠隔装置または遠隔端末から資源またはサービスを要求できる。この点に関しては、A. ピレル、B. ネルソン (A. Birrell and B. Nelson) 著「遠隔手順呼出しの実施 (Implementing Remote Procedure Calls)」(1984年 Vol. 2, No. 1, エーシー・エム・トランザクション・オン・コンピ

ュータ・システムズ(ACM Transaction on Computer Systems)記載)に述べられている。

複数のライセンスサーバが複数の端末に常駐することができる。ライセンスサーバ20は、端末のバックグラウンドモードで作動することによって、該端末を使用するユーザからみて透過的であることが望ましい。また以下で説明するように、ライセンスサーバ20は次のような機能を提供する。

- (1) ライセンシングライプラリからなされるライセンストークンの要求を処理する機能
- (2) いずれのライセンシング単位も使用できない時、データベース18への要求で構成される待ち行列を管理する機能
- (3) データベース18への排他的アクセスのロックを生成する機能
- (4) データベース18内の情報にアクセスする機能

ライセンシングライプラリ24は、ライプラリルーチンの集合であり、アプリケーション26はこれらのルーチンによってライセンスサーバ20にライセンシングサービスを要求する。ライセンスサー

センストークン内のライセンシング情報を確認し、ライセンスが貸出されてよいか否かを決定する。またライセンスアクセスモジュール(LAM)27は、アプリケーションがライセンスサーバ20を介してライセンストークンをデータベース18に戻す前に、該ライセンストークンを符号化している。以下ではライセンスアクセスモジュール27をさらに詳細に説明する。

ライセンス管理ツール21は、ネットワーク管理者がソフトウェアアプリケーションの並行使用に関する管理機能を実行する場合に使用する。ライセンス管理ツール21は、コンピュータネットワークに接続された端末であればいずれの端末でも実行可能である。ライセンス管理ツール21は、ライセンストークンをライセンスサーバ20を介してデータベース18にインストールする場合に主として使用される。ライセンス管理ツール21には次の機能が含まれる。

- (1) ライセンスサーバを起動、停止する機能
- (2) ライセンスサーバの制御するデータベースに

バ20は、ライセンシングライプラリ24からサービスの要求を受信すると、データベース18からライセンストークンを検索して該トークンをライセンシングライプラリ24へと送信する。ライセンシングライプラリ24はアプリケーション26と接続しており、経路28を介してライセンスサーバ20と通信を行なう。かかる通信は、遠隔手順呼出し方式で行なわれる望ましい。ライセンシングライプラリ24における主要なライプラリ呼出しには、アプリケーションがライセンスサーバ20に対して行なうライセンスの要求が含まれる。その他、ライセンスの更新要求と解除要求が重要なライプラリ呼出しの一例である。ライセンストークンを使用して各種のライセンシングサービスの要求を行なう方法を以下に説明する。

ライセンスアクセスモジュール(LAM)27は、ソフトウェアの販売業者が用意するものであって、ライセンストークンの復号化を目的としている。アプリケーション26は、復号化されると、ライセンシングライプラリ内のルーチンを使用してライ

#### アクセスする機能

- (3) ライセンスの使用に関する報告を生成し、印字する機能

アプリケーション26はデータベース18を直接アクセスすることはできない。このアクセスを達成するためには、アプリケーションはライセンシングライプラリ24から経路28を介してライセンスサーバ20宛てにライセンスの要求を行なう。大部分のネットワークライセンシング方式では、ライセンシングライプラリ24とライセンスサーバ20との間に機密保護された伝送路を採用している。しかしながら本発明では、ライセンスアクセスモジュール(LAM)27、ライセンスライプラリ24および複数のライセンストークンを使用してコンピュータネットワークにおけるソフトウェアアプリケーションの不正使用を防いでいる。

再び第2回において、ライセンスサービスバイナリ29は経路30を介してライセンスサーバ20と接続されている。このライセンスサービスバイナリ29は、ネットワークサービスプログラム等の本分

野の公知技術によって呼出される。ライセンスサービスバイング29は、ネットワーク上でサーバとして指定されたすべての端末の位置を検出し、いずれのサーバがいずれのアプリケーションの処理を行なっているかを記録し続ける。ライセンスサービスバイング29は、自己の保有する可用サーバテーブルを参照して各サーバと連絡をとり、サーバの処理している製品のリストを各サーバに要求する。最後に、ライセンスサービスバイング29は、可用ライセンスサーバテーブルの内容と製品リストを、経路32を介して結合ファイル32に蓄込む。第2回において、結合ファイル32は経路33を介してライセンシングライプラリ24に接続されている。アプリケーション28は結合ファイル32に問い合わせを行ない、いずれのライセンスサーバが該アプリケーションのライセンスの要求を処理できるかを確認する。ライセンス生成ツール34は、ネットワーク管理者に伝達されるライセンストークンを生成するためにソフトウェア販売業者が使用する。ネットワーク管理者は、ライセンストークンを受

信すると、ライセンスサーバ20を介してライセンス管理ツール21によって該トークンをデータベース48へとインストールする。

### 11. ライセンストークン

本発明の好ましい実施例を使用してコンピュータネットワークでライセンストークンを生成する方法を第3図を参照しながら説明する。第3図において、コンピュータネットワーク38は、ライセンス管理ツール39と1個のライセンスサーバ44とに接続されている。ライセンスサーバ44はデータベース45と通信を行なう。アプリケーション41、42、43はライセンスサーバ44からライセンシングサービスを要求している。顧客が任意のアプリケーション(例: 研究開発部門用のCAD/CAMプログラム)のライセンスを購入すると、ソフトウェア販売業者はライセンス生成ツールによってライセンストークンを生成して、該トークンを顧客のネットワーク管理者に渡す。ライセンストークンは、ソフトウェアアプリケーションを使用するためのライ

センスを表現する特定のビットパターンすなわちパケットである。ネットワーク管理者は、ライセンストークン46を、ライセンス管理ツール38を使用してライセンスサーバのデータベースへとインストールする。トークンリング方式で端末から端末へと転送されるトークンと異なり、本実施例のライセンストークンは、一定の時間中ライセンスサーバとライセンシングライプラリとの間でのみ転送される。かかる一定期間は、ライセンストークンがライセンスサーバから貸出される時間の長さに相当する。現在、ライセンストークンはアプリケーションに対して10秒以内に転送され、該トークンはこれを貸出したライセンスサーバへと最短時間で返送される。ライセンストークン46は、販売業者独自のフォーマットで暗号化された情報を含んでいる。かかる情報には、販売業者の識別記号、製品番号、バージョン番号、ライセンストークンに対して購入されたライセンス単位の数などがある。ライセンス単位は、コンピュータネットワークに接続された端末に加重されたライ

センスに対応している。たとえば、任意のソフトウェアアプリケーションを使用する場合、強力なワークステーションは平均的なパソコンより多くのライセンス単位を必要とすることになる。

ソフトウェア販売業者は、ライセンス生成ツール40を使用してライセンストークンを生成する。経路47は、顧客のサイトでライセンストークン46'がどのようにライセンス管理ツール38へと送られるかを示している。ここでは、システム管理者はライセンストークン46'をライセンストークン46としてライセンスサーバ44のライセンスデータベース45へとインストールする。経路48は、ライセンストークン46'がライセンストークン46としてライセンス管理ツール38からライセンスサーバ44へと転送される様子を示している。これにより、ライセンスサーバ44においては、トークン46に該当するアプリケーションを使用するライセンスの要求や、データベース45で示される他のアプリケーションを使用するライセンスの要求を、アプリケーション41、42、43から受け付ける用意が

完了する。

各ネットワークは複数のライセンスサーバを保有可能であり、また各ライセンスサーバは、そのデータベース内に複数のソフトウェアアプリケーションに対応する複数のライセンストークンを含んでいてよいことは明らかである。再び第3図において、アプリケーションA 41が10秒未満でライセンストークン48を要求してこれを借り出すと、アプリケーションA 41がライセンスをライセンストークン46から借り出している期間中にアプリケーションB 42、C 43がこれに対する要求を行なっても、ライセンストークン46を借り出すことが出来ない。これはデータベースインクフェース18がロック情報を提供しているためである。このように、ネットワーク38でライセンスの並行使用を達成するためには、ネットワーク管理者が2個以上のライセンスサーバをインストールすることが望ましい。ライセンスサーバのクラッシュから回復するための労力を軽減するためには、システム管理者が各アプリケーションのライセンス単位を戦略的

に配置された複数のライセンスサーバに分散しておくことが望ましい。たとえば、ネットワークに4個のライセンスサーバがある場合、使用頻度の高いあるアプリケーションに対して20個のライセンス単位があるとすれば、ネットワーク管理者は4個のライセンストークンの各々に5個のライセンス単位を配分することが好ましい。1個のライセンスサーバがクラッシュした場合、もしくは該サーバのライセンストークンが貸出されている場合、他の3個のライセンスサーバが他のアプリケーションに対してライセンシング処理を行なうことができる。

第4a図は、ライセンストークンを使用してライセンスを要求する方法を示している。第4a図において、ネットワーク50がアプリケーション52、54、58にそれぞれ結合されている。アプリケーション58は、ステップ59においてライセンスサーバ58からライセンストークンを要求し、該要求は受け付けられる。ステップ60において、ライセンストークンはアプリケーション58へと伝達される。続い

てアプリケーション58は、ステップ61においてライセンストークンをライセンスサーバ58に送り返す。第4a図に示されるようなライセンストークンによるライセンス要求機能に加えて、ライセンシング処理の他の重要な段階でもライセンストークンが使用される。たとえば、ユーザが当初割当てられた時間を超過してアプリケーションを使用したい場合がある。この場合、第4b図に示されるように、アプリケーション68は、ライセンストークン72を使用してライセンスサーバ70に対してライセンス更新要求を発行する。同様に、第4c図に示されるように、ユーザはアプリケーションがライセンス単位を必要としなくなった時点でライセンス解除要求83を発行する。このように、ユーザはステップ85において、更新されたライセンストークンをライセンスサーバ82に戻すことによってライセンストークン84を更新する。

### III. ライセンスアクセスモジュール

第2図において、ライセンスアクセスモジュール

ル(LAM)27は、アプリケーション28とライセンシングライブラリ24とに結合されて実行可能コードを形成する。該コードは、ソフトウェア版充葉者が顧客に出荷するコードである。ライセンスアクセスモジュール27は、暗号化されたライセンストークンがライセンスサーバとライセンシングライブラリ24との間で転送される時点でその復号化と符号化を行なう。このように、アプリケーションの不正使用に対する機密保護のレベルは、ライセンスアクセスモジュールの機密保護の度合いに大きく依存する。

従来のネットワークライセンシングの方法では、公共／個人キー暗号化方式を使用して重要な情報を符号化している。この方式は、ライセンスサーバが安全な環境内にある場合には効果的である。しかしながら、顧客はライセンスサーバも含めたネットワークの全端末に同じようにアクセスすることができる。ユーザがライセンスサーバの個人キーを解読すれば、ライセンシング方式の機密保護は破られてしまう。無資格のユーザがサーバの

個人キーをいったん特定すれば、ライセンスサーバに関する他のすべての重要な情報が解読されてしまう。仮にすべてのライセンスサーバが同一のキーを使用している（これがしばしば行なわれている）とすれば、すべてのライセンスサーバの扱うアプリケーションのすべての権限保護が破られてしまう。

ライセンスアクセスモジュール27は、ライセンストークンを販売業者指定のフォーマットからライセンシングライブライアリ24で使用可能なフォーマットへとまず変換する。ライセンスアクセスモジュールによるこの変換は2個のモジュールによって行なわれる。第一のモジュールは、ライセンストークンを販売業者指定のフォーマットからライセンシングライブライアリのフォーマットへと変換する。第二のモジュールは、更新されたライセンストークンをライセンシングライブライアリのフォーマットから販売業者指定のフォーマットへと変換する。第二のモジュールは、ライセンシングライブライアリがライセンス

トークン内の情報を更新するたびごとに呼び出される。

ライセンシングライブライアリマットに変換されたライセンストークンの受信後、ライセンシングライブライアリはライセンスの正当性を確認するためのルーチンを呼び出す。該確認はトークン内に格納された次のようなライセンス情報を検討することによって行なわれる。

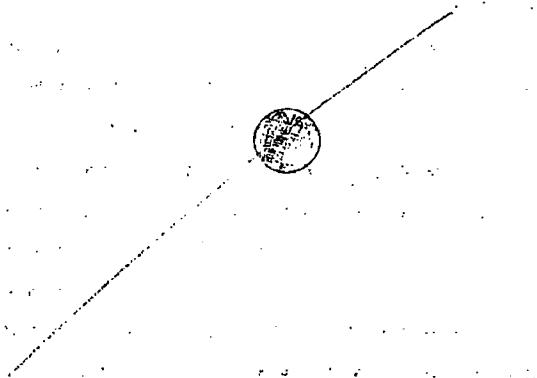
- (1) フラグ
- (2) 保守契約の日付
- (3) ホストの名称と定義域
- (4) 製品名
- (5) ホストの識別番号
- (6) ライセンスの返し番号
- (7) ライセンスの失効の日付

以上の情報がアプリケーションの保有する情報と比較される。これらの情報が合致すれば、ライセンスの確認が完了する。確認過程の終了後、ライセンシングライブライアリ内でルーチンが起動される。このルーチンは、ライセンストークン内のラ

イセンス単位数から貸出中のライセンス単位の数を減算してライセンスを貸出す。

復号化ルーチンと符号化ルーチンとによって、ソフトウェア販売業者は、顧客のサイトに常駐するライセンスを不正使用から守るための独自の権限保護機構を構築することができる。

以下に、ライセンシングライブライアリとライセンスアクセスモジュールを使用するアプリケーションであって、C言語で記述されたアプリケーションの例を示す。



```

5 #define LIC_RENEWAL_TIME {60}           /*set renewal time for this session/
6 #define EST_LIC_RENEWAL_TIME (LIC_RENEWAL_TIME * .9)
7
8     /*odd, status);
9     return;
10 }
11
12 int
13 alarm_intr ()
14 {
15     status = NL_confirm (req_handle,
16                         LIC_RENEWAL_TIME, NULL);
17
18     /* Verify vendor private information
19     */
20
21     if (status != NL_NO_ERROR)
22     {
23         ifprint (sider, "nl_confirm failed - error = "
24                 "%d\n", status);
25     }
26
27     /* Renew licensing unit with
28      licensing server*/
29
30     if (status == NL_NO_ERROR)
31     {
32         /*otherwise, error
33          message/
34          message/
35          successful license
36          renewal*/
37
38         ifprint (sider, "nl_confirm failed - error = "
39                 "%d\n", status);
40
41         /* nl_confirm failed - error = NL_NO_ERROR
42          */
43
44         /* Renew licensing unit with
45          licensing server*/
46
47         ifprint (sider, "nl_renewal failed - error = "
48                 "%d\n", status);
49
50     }
51
52     /*odd, status);
53     return;
54 }
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1196
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1388
1389
1390
1391
1392
1393
1394
1395
1396
1396
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1496
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1596
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1696
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1796
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1896
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2279
2280
2281
2282
2283
2284
2285
2286

```

ヨンと結合されたライセンシングライブラリによってライセンストークンの確認手順が実行される。

システムの権限保護のレベルはライセンスアクセスモジュールによって決定されるため、ソフトウェア販売業者は該モジュールを任意の難易度で生成することができる。とりわけ、ソフトウェア販売業者は、暗号ルーチンの一部としていずれの暗号方式でも採用することができる。権限保持機構が解読されて暗号が知られてしまった場合、ソフトウェア販売業者は、新しいライセンスアクセスモジュールを使用して製品の新しいバージョンをリリースするだけで状況を改善することができる。

本発明は、前述のように第1-4図と付録1を参照しながら具体的に説明され、とりわけコンピュータネットワーク環境におけるソフトウェアアプリケーションの不正使用を防ぐ方法の実施について具体的に説明されたが、かかる説明は例であって本発明を限定するものではない。また本発明による方法は、コンピュータネットワーク環境で実行

されるいかなるアプリケーションに用いても効果的であることは明白である。さらに、以上の説明から本発明の精神と範囲とにそむくことなくさまざまな変更や修正を実行できることは当該分野の有能な技術者によって了解されよう。

明細書の序(内容に変更なし)  
1-1 表

```
/*
 * 0.0(nl_lic.h, 1.21.09/09/13
 * Copyright (c) 1998 by Sun Microsystems, Inc.
 */
*****  

* Module: nl_lic.h
*****  

* Description:
 * Contains the license access module. Uses XDR to keep the license
 * token in a machine independent form. Uses the des_crypt() routines
 * to encrypt/decrypt the licensing information.
*****  

* Functions:
 * nl_decode()           - Decodes a vendor's license token
 * nl_encode()           - Encodes a vendor's license token
*****  

* Notes:
*****  

* *****  

* Includes:
 * <rpr/rpr.h>
 * <string.h>
 * <memory.h>
 * <nl_types.h>
 * <nl_prot_limits.h>
 * <nl_request.h>
 * <nl_token.h>
 * <nl_vendor.h>
 * <nl_liconfig.h>
 * <nl_error.h>
*****  

* *****  

* Module Local Definitions
*****  

* *****  

* Warning: This could be a potential security threat, leaving a unencoded
 * token lying around in memory.
 * /  

* static NL_license _nl_lic_tok;
*****  

* *****  

* Function Declarations
*****  

* /  

* void nl_lic();
 * static int _nl_decode();
 * static int _nl_encode();
*****  

* /
```

明細書の添書(内容に変更なし)

### 明細書の添書(内容に変更なし)

### 明細書の添書(内容に変更なし)

明細書の添書(内容に変更なし)

```

NL_SERIAL_NUM_SIZE)
tok_p->expire_secs = nl_llc_tok_expire_secs;
tok_p->parent_secs = nl_llc_tok_main_secs;
tok_p->total_units = nl_llc_tok_total_units;
tok_p->available_units = nl_llc_tok_avail_units;
tok_p->available = nl_llc_tok_avail;
tok_p->strncpy(tok_p->domain, nl_llc_tok_domain,
NL_DOMAIN_NAME_SIZE);
tok_p->host = nl_llc_tok_host;
tok_p->flag = nl_llc_tok_flag;
tok_p->num_clients = nl_llc_tok_num_clients;

/* Give the user a copy of the vendor private data if they
 * want it.
 */
vend_size = MIN(nl_llc_tok.vendor_priv_size,
NL_VEND_PRIV_SIZE);
if (vend_size > 0 & tok_p->vendor_priv_p != NULL)
{
    tok_p->vendor_priv_p->length = vend_size;
    (void) memop(tok_p->vendor_priv_p->data_p,
nl_llc_tok.vendor_priv,
NL_VENDOR_SIZE, tok_p->vendor_priv_p->length));
}

tmp_clnt_entry_p = (NL_clnt_entry *) sizec(NL_clnt_entry);
malloc((tok_p->num_clients)* sizec(NL_clnt_entry));
*clnt_entry_p = tmp_clnt_entry_p;
for (i=0; i < tok_p->num_clients; i++)
{
    NL_usage_entry_taddr, usage_entry);
    tmp_clnt_entry_p->req.handle = usage_entry.req.handle;
    tmp_clnt_entry_p->req.handle = usage_entry.req.handle;
    tmp_clnt_entry_p->unit = usage_entry.unit;
    tmp_clnt_entry_p->renew_secs = usage_entry.renew_secs;
    tmp_clnt_entry_p->clnt_secs = usage_entry.client_secs;
    tmp_clnt_entry_p->pid = usage_entry.pid;
}

else
{
    status = NL_E_DECODE_PATH;
}

xdr_destroy(xaddr);
return (status);
} /* nl_encode */

nl_encode(tok_p->olnt_entry_p, en_tok_p, en_length_p)
NL_token *tok;
NL_clnt_entry *olnt_entry;
NL_clnt_entry *en_tok_p;
int en_length_p;
{
    NL_usage_entry usage_entry;
    NL_clnt_entry *clnt_entry_p;
    int i;
    int tok_len;
    int tok_len2;
    int int;
    int status;
    status = NL_NO_ERROR;

    /* Init an XDR stream
    */
xdrmem_create(xdr, en_tok_p, (u_int)NL_MAX_TOKEN_SIZE, XDR_ENCODE);
{
    /* If for some reason the license token that we will encode is
     * not the same as the previous one decoded (in our module
     * or local global) then we copy the information in.
     */
    if (memcmp(nl_llc_tok.serial_num, tok_p->serial_num,
NL_SERIAL_NUM_SIZE) != 0)
        nl_llc_tok.magic = MY_VENDOR_MAGIC;
}
}

```

## 明細書の添付(内容に変更なし)

- Quick check here for the first pass of this loop. We find out how many bytes it took to encode 1 user entry. We then estimate the total number of bytes required and make sure that we will not exceed NL\_MAX\_TOKEN\_SIZE.

```

if (i == 0)
{
    tok_len2 = (int)xdr_getpos(xdrs) -
        tok_len1;
    tok_len2 -= tok_p->num_clients;
    if ((tok_len1+tok_len2) >
        NL_MAX_SN_TOKEN_SIZE)
    {
        i = tok_p->num_clients;
        status = NL_E_ENCODE_FAIL;
    }
}

free ((char *)olent_entry_p);
sen_length_p = (int)xdr_getpos(xdrs);
xdr_destroy(xdrs);

return (status);
/* nl_encode */

```

```

nl_llic Tok.Vendor.Id = tok_p->Vendor.Id
NL_PROD_NUM_SIZE];
NL_PROD_NUM_SIZE];
NL_SERIAL_NUM_SIZE];
NL_SERIAL_NUM_SIZE];
NL_LIC_Tok.Expire_Seconds = tok_p->Expire_Seconds;
NL_LIC_Tok.Maint_Seconds = tok_p->Maint_Seconds;
NL_DOMAIN_NAME_SIZE];
NL_DOMAIN_NAME_SIZE];
NL_LIC_Tok.HostId = tok_p->HostId;
NL_LIC_Tok.Flag = tok_p->Flag;
If (tok_p->Vendor.Priv_p != NULL) &&
    tok_p->Vendor.Priv_p->data_p != NULL)
{
    _nl_llic Tok.Vendor.Priv_size =
        MIN(tok_p->Vendor.Priv_p->length,
            NL_VEND_PRIV_SIZE);
    (void) memcpy(_nl_llic Tok.Vendor.Priv,
        tok_p->Vendor.Priv.p->data_p,
        _nl_llic Tok.Vendor.Priv_size);
}
else
{
    _nl_llic Tok.Total_Units = tok_p->Total_Units;
    _nl_llic Tok.Avail_Units = tok_p->Avail_Units;
    _nl_llic Tok.Num_Clients = tok_p->Num_Clients;
    if (xdr_NL_Licensees(xdrs, &nl_llic Tok) == FALSE)
    {
        xdr_destroy(xdrs);
        return (NL_E_ENCODE_FAIL);
    }
    if (olent_entry_p != NULL)
    {
        tmp_olent_entry_p = (NL_Olent_Entry *) malloc((NL_Olent_Entry_size) *
            sizeof(NL_Olent_Entry));
        tmp_olent_entry_p->olent_entry_p->Handle =
            tok_len1 - (int)adr_Setups((xdr));
        for (i=0; i < tok_p->num_clients; i++)
            tmp_olent_entry_p->clients[i] = tok_p->clients[i];
        usage_entry_req.Handle = tmp_olent_entry_p->req_Handle;
        usage_entry_req.Units = tmp_olent_entry_p->units;
        usage_entry_req.Expire_Seconds = tmp_olent_entry_p->expire_Seconds;
        usage_entry_req.Maint_Seconds = tmp_olent_entry_p->maint_Seconds;
        usage_entry_req.Id = tmp_olent_entry_p->id;
        xdr_NL_usage_Entry(xdr, usage_entry);
    }
}

```

## 明細書の添付(内容に変更なし)

## 明細書の添付(内容に変更なし)

## 4. 図面の簡単な説明

第1図は本発明を使用したネットワーク環境を示す図、第2図は本発明の好ましい実施例を使用したネットワークライセンシング方式の構造を示す図、第3図はライセンストークンを本発明の好ましい実施例にインストールした様子を示す図、第4 a図は本発明の好ましい実施例においてライセンストークンを使用してライセンスサーバからライセンスを要求する様子を示す図、第4 b図は本発明の好ましい実施例においてライセンストークンを使用してライセンスサーバからのライセンスを更新する様子を示す図、第4 c図は本発明の好ましい実施例においてライセンストークンを使用してライセンスサーバからのライセンスを解除する様子を示す図である。

21 \* \* \* ライセンス管理ツール、24 \* \* \* ライセンシングライブリ、34 \* \* \* ライセンス生成ツール、32 \* \* \* 結合ファイル。

特許出願人 サン・マイクロシステムズ・  
インコーポレーテッド

代理人 山川政樹

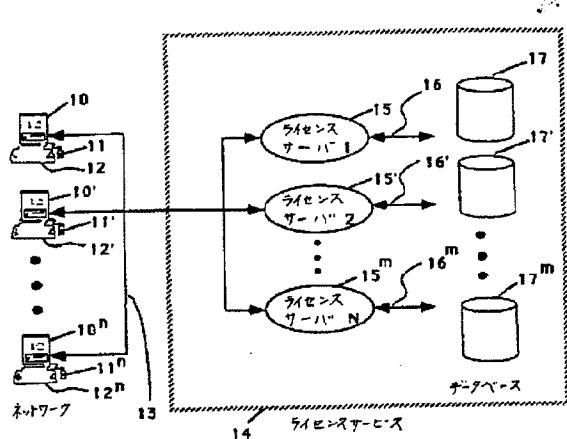


FIG. 1

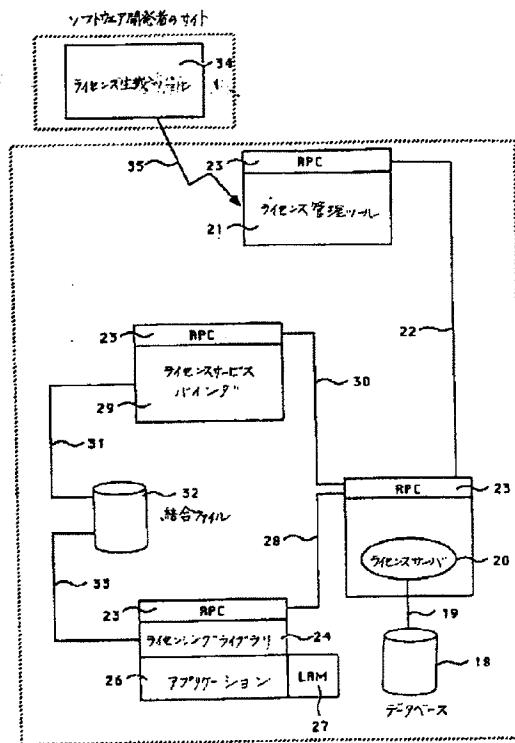


FIG. 2

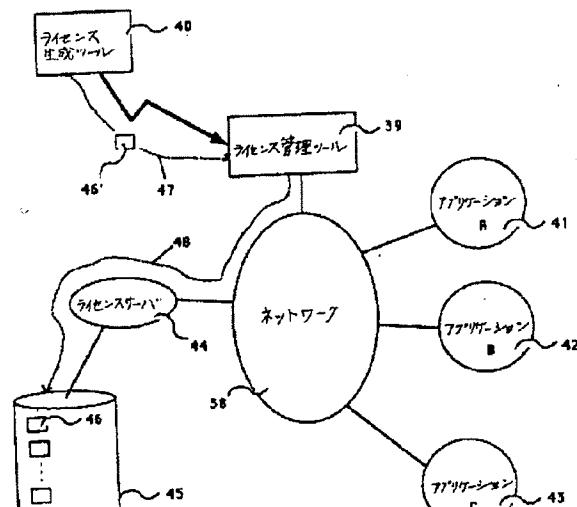


FIG. 3

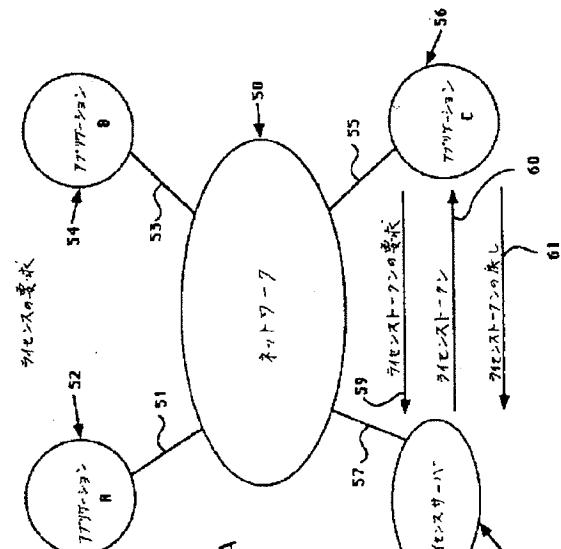


FIG. 4A

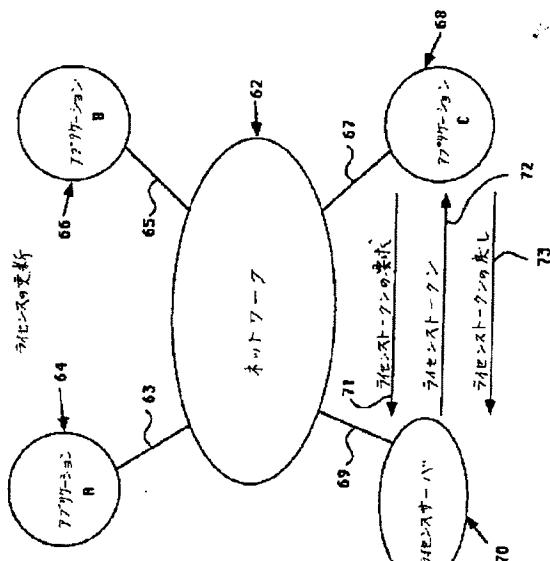


FIG. 4B

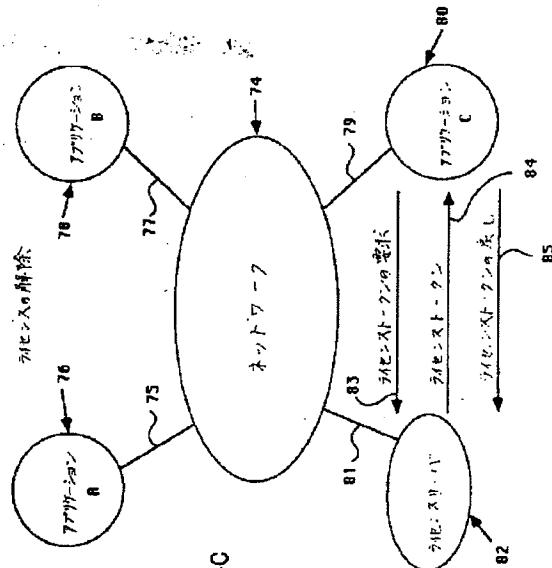


FIG. 4C

## 手 続 補 正 書 (方式)

平成 3年 5月16日

特許庁長官殿

## 1. 事件の表示

平成2年特許願第263242号

## 2. 発明の名称

コンピュータネットワークシステム内の  
ソフトウェアの不正使用を防ぐ方法

## 3. 補正をする者

事件との関係 特許出願人

名称(氏名) サン・マイクロシステムズ・  
インコーポレーテッド

## 4. 代理人

居所 東京都千代田区永田町2丁目4番2号  
秀和溜池ビル8階  
山川国際特許事務所内  
電話 (3580) 0961 (代表)

氏名 (6462) 弁理士 山川政樹



## 5. 補正命令の日付 平成3年4月16日

## 6. 補正の対象

(1) 明細書 41P~47P

## 7. 補正の内容

(1) 明細書の净書(内容に変更なし)

以上  
3.5.16  
3.5.16

## 手 続 補 正 書 (方式)

平成 年 月 日  
3. 11. 7

特許庁長官 殿

## 1. 事件の表示 特願平2-263242号

## 2. 発明の名称

コンピュータネットワークシステム内の  
ソフトウェアの不正使用を防ぐ方法

## 3. 補正をする者

事件との関係 特許出願人

名称 サン・マイクロシステムズ・  
インコーポレーテッド

## 4. 代理人

居所 東京都千代田区永田町2丁目4番2号  
秀和溜池ビル8階  
山川国際特許事務所内  
電話 (3580) 0961 (代表)

氏名 (6462) 弁理士 山川政樹



## 5. 補正命令の日付 平成3年10月22日

## 6. 補正の対象

明細書(第48頁)

## 7. 補正の内容

明細書の净書(内容に変更なし)

3.11.7  
3.11.7  
3.11.7

【公報種別】特許法第17条の2の規定による補正の掲載

【部門区分】第6部門第3区分

【発行日】平成11年(1999)4月30日

【公開番号】特開平4-100148

【公開日】平成4年(1992)4月2日

【年通号数】公開特許公報4-1002

【出願番号】特願平2-263242

【国際特許分類第6版】

G06F 9/06 550

15/00 330

【F I】

G06F 9/06 550 Z

15/00 330 Z

特許請求の範囲

平成 年 月 日  
9.9.4

特許庁長官殿

1. 事件の表示 平成2年特許第263242号

2. 補正をする者

事件との関係 特許出願人

名称 サン・マイクロシステムズ・インコーポレーテッド

3. 代理人

識別番号 100064621

弁理士・専門  
氏名 山川政樹

電話番号 03-3580-0961

4. 補正により増加する請求項の数 : 6

5. 補正対象書類名 明細書

6. 補正対象項目名 特許請求の範囲

7. 補正の内容 特許請求の範囲を、下記の通り補正する。

1. 少なくとも一つのアプリケーション・プログラムを実行する少なくとも一つのコンピュータを含むコンピュータ・ネットワークにおいて、前記アプリケーション・プログラムの許可の実行に対するネットワーク全範囲にわたる保護をさらに増強する、アプリケーション・プログラムをライセンスするための方法であって、

前記アプリケーション・プログラムのライセンシング情報を含み、暗号化された書式で記憶される第1ライセンス・トークンを、第1ライセンス・サーバによって第1データベースに記憶するステップと、

前記アプリケーション・プログラムを実行のためにライセンスするため、ライセンシング・ライブリのライセンシング・ルーチンによって前記第1ライセンス・サーバに要求されるに応じて、第1ライセンス・サーバによって前記アプリケーション・プログラム用の前記の記憶された第1ライセンス・トークンを検索するステップと、

検索された前記第1ライセンス・トークンが前記アプリケーション・プログラムと統合されたアプリケーション・プログラム専用のライセンス・アクセス・モジュールによって暗号解読された後、検索された前記第1ライセンス・トークンに記憶されている前記ライセンシング情報を、前記ライセンシング・ルーチンと共に前記アプリケーション・プログラムによって維持されるライセンシング情報を交き合わせて検索するステップと、

前記ライセンシング・ルーチンによる前記検索に応じて、実行のため前記アプリケーション・プログラムをライセンスするステップと、

前記ライセンシング・ルーチンによる前記ライセンシングに応じて、検索された前記第1ライセンス・トークンを更新するステップと、

更新された前記第1ライセンス・トークンが前記アプリケーション・プログラムと統合された前記アプリケーション・プログラム専用のライセンス・アクセス・モジュールによって再暗号化された後、更新済みの前記第1ライセンス・トークンを、前

記第1データベースに記憶するため前記ライセンシング・ルーチンによって前記第1ライセンス・サーバに戻すステップとを含む方法。

2. 前記第1ライセンス・トークンを前記アリケーション・プログラムの発生から受け取り、前記第1ライセンス・トークンを前記第1データベースに記憶するため前記第1ライセンス・サーバに供給するステップをさらに含み、

前記第1ライセンス・トークンは、前記アリケーション・プログラムの配布媒体から受信され、暗号化された書式でライセンス管理ツールによって前記第1ライセンス・サーバに供給されることを特徴とする、  
請求項1に記載の方法。

3. 前記アリケーション・プログラムの配布媒体からの、前記アリケーション・プログラム特有のライセンス・アクセス・モジュールと統合され、前記コンピュータにインストールされた、前記アリケーション・プログラムを受信するステップをさらに含み、

前記アリケーション・プログラム特有のライセンス・アクセス・モジュールが、前記アリケーション・プログラムと統合され、前記アリケーション・プログラムの発生によって前記配布媒体に図かれていることを特徴とする、  
請求項1に記載の方法。

4. 前記アリケーション・プログラムを実行するためライセンス不許可にするため、ライセンシング・ライブリのライセンス不許可ルーチンによって前記第1ライセンス・サーバに要求されるに応じて、前記アリケーション・プログラム用の前記の記憶された第1ライセンス・トークンを前記第1ライセンス・サーバによって検査するステップと、

検査された前記第1ライセンス・トークンが前記アリケーション・プログラムと統合されたアリケーション・プログラム特有のライセンス・アクセス・モジュールによって暗号解読された後に、検査された前記第1ライセンス・トークンに記憶されている前記ライセンシング情報を、前記ライセンス不許可ルーチンと共に前記アリケーション・プログラムによって維持されるライセンシング情報をと交換させて確認するステップと、

前記ライセンス不許可ルーチンによる前記確認に応じて、実行のため前記アリケーション・プログラムをライセンス不許可にするステップと、

前記ライセンス不許可ルーチンによる前記ライセンス不許可に応じて、後者された前記第1ライセンス・トークンを返却するステップと、

更新済みの前記第1ライセンス・トークンが前記アリケーション・プログラムと統合された前記アリケーション・プログラム特有のライセンスモジュールによって再暗号化された後、更新済みの前記第1ライセンス・トークンを、前記第1データベースに記憶するために前記ライセンス不許可ルーチンによって前記第1ライセンス・サーバに戻すステップとをさらに含む請求項1に記載の方法。

5. 前記アリケーション・プログラムのライセンシング情報を含み、暗号化された書式で記憶される第2ライセンス・トークンを、第2ライセンス・サーバによって第2データベースに記憶するステップと、

前記ライセンシング・ライブリが組じ込みファイルにアクセスして、前記第1ライセンス・サーバを選択し前記第1ライセンス・トークンを検査するように、前記の第1ライセンス・サーバと第2ライセンス・サーバを識別する第1ライセンス・サーバ識別子と第2ライセンス・サーバ識別子を組じ込みファイルに記憶するステップと、

前記の第1ライセンス・サーバと第2ライセンス・サーバを識別して、前記の第1ライセンス・サーバ識別子と第2ライセンス・サーバ識別子をライセンス・サービス・バインディングによって前記の組じ込みファイルに記憶するステップとをさらに含む請求項1に記載の方法。

6. 少なくとも一つのアリケーション・プログラムを実行する少なくとも一つのコンピュータを含むコンピュータ・ネットワークにおいて、前記アリケーション・プログラムの無許可の実行に対するネットワーク全範囲にわたる保護をさらに強化する、アリケーション・プログラムをライセンスするための装置であって、

前記アリケーション・プログラムのライセンシング情報を含み、暗号化された書式で記憶される第1ライセンス・トークンを記憶するための第1データベース

とを含む第1データベース手段と、

前記第1ライセンス・トークンを前記第1データベースに記憶し、かつ記憶された前記第1ライセンス・トークンを前記第1データベースから検査するための第1ライセンス・サーバを含む、前記第1コンピュータと前記第1データベース手段とに結合された第1ライセンス・サーバ手段と、

前記アリケーション・プログラム用の前記第1ライセンス・トークンを検査するよう前記第1ライセンス・サーバに要求し、検査された前記第1ライセンス・トークンが前記アリケーション・プログラムによって暗号解読された後に、検査された前記第1ライセンス・トークンに記憶されている前記ライセンシング情報を、前記アリケーション・プログラムによって維持されたライセンシング情報をと交換させて確認し、前記暗号によって前記アリケーション・プログラムを実行のためにライセンスし、前記ライセンシングに応じて、検査された前記第1ライセンス・トークンを更新し、かつ更新済みの前記第1ライセンス・トークンを前記アリケーション・プログラムによって再暗号化された後に、更新済みの前記第1ライセンス・トークンが前記アリケーション・プログラムによって再暗号化された後に、前記アリケーション・プログラムによって再暗号化された後に、前記アリケーション・ライセンシング・ルーチンを含むライセンシング・ライブリと、

前記アリケーション・プログラム特有の、前記アリケーション・プログラムと統合された、検査された前記第1ライセンス・トークンを暗号解読し、かつ更新済みの前記第1ライセンス・トークンを再暗号化するための、前記コンピュータによって実行されるライセンス・アクセス・モジュールとを含む装置。

7. 前記アリケーション・プログラムの配布媒体から前記第1ライセンス・トークンを最初に受け取り、前記第1ライセンス・トークンを前記第1データベースに記憶するため前記第1ライセンス・サーバに供給するための、前記第1ライセンス・サーバ手段に結合されたライセンス管理ツール手段を含み、前記第1ライセンス・トークンが前記ライセンス管理ツールによって記憶され、暗号化された書式で前記第1ライセンス・サーバに供給されることを特徴とする、請求項6に記載の装置。

8. 前記アリケーション・プログラムのライセンシング情報を含み、暗号化された書式で記憶されている、第2ライセンス・トークンを記憶するための第2データベースを含む第2データベース手段と、

前記第2ライセンス・トークンを前記第2データベースに記憶し、かつ記憶された前記第2ライセンス・トークンを前記第2データベースから検査するための第2ライセンス・サーバを含む、前記コンピュータと前記第2データベース手段とに結合された第2ライセンス・サーバ手段と、

前記ライセンシング・ライブリが組じ込みファイルにアクセスして、前記第1ライセンス・サーバを選択し前記第1ライセンス・トークンを検査するように、前記の第1ライセンス・サーバと第2ライセンス・サーバを識別する第1ライセンス・サーバ識別子と第2ライセンス・サーバ識別子を記憶するための、組じ込みファイルを含む前記のコンピュータに結合された組じ込みファイル手段と、

前記第1ライセンス・サーバ手段と、前記第2ライセンス・サーバ手段と、前記組じ込みファイル手段とに結合され、前記の第1ライセンス・サーバと第2ライセンス・サーバを識別して、前記の第1ライセンス・サーバ識別子と第2ライセンス・サーバ識別子を前記組じ込みファイルに記憶するためのライセンス・サービス・バインディングを含む、ライセンス組じ込み手段とを含む請求項6に記載の装置。